Citizen-driven smart e-government: a personal dashboard for using web services and open data

GIOVANNI AIELLO¹, GERARDO CANFORA², EUGENIO ZIMEO²

¹Engineering Ingegneria Informatica
²Department of Engineering – University of Sannio
Service oriented architectures play an important role in the modernization process of Public Administration

- Goal: reducing costs and improving services provided to citizens

State of the art: in the last 10 years, large efforts have been devoted to enable interoperability among PAs

- eGov envelope, gateway components, communication paradigms, SLA definition, semantics, etc.
However, to date, the deployment of public services and their use is still poor! **WHY?**

By focusing mainly on the backend, citizens have been excluded from the “PA servicing process”
- They are only consumers of services, while PA is the **owner** of ideas, data and services
- Services are exposed to citizens through custom Web applications

**Open Data contribute to introduce a significant paradigm shift from citizen-centric to citizen-driven e-Government**
- PA is **owner only** of data
- Data are usable for the development of new ideas and services of social interest

We think that by exploiting a peer-to-peer development model, citizens and organizations having access to open data can create an effective ecosystem of social services
Mashup: a web application that uses content from more than one source to create a single new service displayed in a single graphical interface (Widget)

- E.g. user could combine the addresses of libraries branches with a Google map to create a map mashup
Mashup discovery through contextual info

MyOpenGov project - activity 2.1

Bar section for:
- user’s profile access
- widget store access
- user logout

Connection to Social Networks accounts

Metawidget of:
- recommended widgets
- all widgets in the store

Metawidget of recommendend widgets from friends

Metawidget for widgets search based on text

Metawidget for widgets developers notification

Widget used by the user

Buttons for like and tagging
System architecture

- **Composed by:**
  - **Dashboard,** for implicit and explicit discovery of service compositions and their sharing
  - **Recommender,** for producing personalized suggestions to dashboard users
  - **Extractor,** for extracting, from social networks, data (profile, interactions and preferences) related to registered users
Service discovery: Recommender

- Different algorithms for implicit discovery, based on similarities between
  - Mashups
  - Mashups and social network preferences
  - Users

- Explicit discovery based on
  - Full-text search
  - Tag-based search

- Collaborative tagging and recommendation
Mashups similarity

- The algorithm suggests other widgets similar to the one in context
  - Triggered when a dashboard user selects a widget for his/her personal space

- Similarities evaluated with Lucene text engine
  - By analyzing title, description and service URIs of each mashup and by applying IR algebraic models

- Similarity can be also inferred by common tags?
Users similarity

- Based on social interactions (e.g. friendships)
  - low reliability

- Based on social activities and social topology
  - e.g. user a -> user b similarity (asymmetric similarity)
    - #likes on pages (and posts) of user b
    - #common friends
    - degree centrality

\[
a \ast \text{sigmoid}(\#\text{likes}_{a,b}) + (1-a) \ast \frac{\#\text{common\_friends}_{a,b}}{\text{degree}_a}
\]

- higher reliability
Mashups and SNs Preferences similarity

- The algorithm suggests widgets that are similar to the preferences declared on social networks
  - Triggered when a dashboard user connects his/her account to a social network one
- Same similarities evaluation
- LinkedIn *skills* used for notifying widgets developers
The algorithm collects the tags belonging to mashups whose similarity score is higher than $x$ (average, median or fixed threshold).

Each tag recommendation score is based on:
- #times the tag is discovered
- #times the tag is used by the user
- user similarities
  - social (friends) and users tagging similarities based on common tags used

Outputs:
- Tags recommended for a mashup in context
- Friends and users tagging similarities
- Tagging can be used to infer mashup similarities

Triggered when a user wishes to tag a mashup
SN Data Extractor

- Subsystem used to get social networks data
- Based on Oauth and developed by using **Spring (MVC and Social)** to mask providers (graph) APIs (Facebook, LinkedIn, etc...)

**Extracts:**
- Personal basic profile information (i.e. first name, last name, etc..,)
- Social preferences (i.e. pages liked)
- Skills (i.e. what the user declare to be able to do)
THANK YOU

giovanni.aiello@eng.it
gerardo.canfora@unisannio.it
eugenio.zimeo@unisannio.it